Expert System for Educational Content Selection Based on Standards

Santiago Pérez de la Cámara, Edgar Rubión Soler, Alicia Fernández del Viso Torre, Eva Vázquez de Prada, Carlos Rebate Sánchez
{sperezd, erubion, afernandezde, evazquezdeprada, crebate}@indra.es
eInclusion Unit, Indra Software Labs
C/ Acanto 11, 28045 Madrid (Spain)

Abstract. The most common approach to content selection for educational purposes is often focused on how to display these contents in the variety of devices that the user could have (PC, PDA, mobiles, etc.); neither user needs nor preferences are taken into account, nor the desired features for accessible content. This paper depicts the standard based Expert System that is being developed on the framework of the European project EU4ALL [European Unified Approach for Assisted Life Long Learning - IST-FP6-034778]. The goal of this component is to select the content that provides best adaptation to the user. The first prototype of this component has been developed following a SOA (Service Oriented Architecture) approach so it can be easily integrated with existing Virtual Learning Environment.

Keywords. Content selection, standards, accessibility, Expert Systems, SOA

Introduction

Nowadays many educational institutions open their virtual classes to students with different disability levels. The main challenge that students with disabilities face is to access education resources that are appropriate for them. Educational resources are usually the same for every student since no specific adaptation is provided to adapt the resource to the needs of the students. As a result, students cannot take full advantage of the learning experience that virtual learning environments offer. They are several works about the study of the content personalization as we can see in [1], [2], [3], [4] and [5].

In this paper we depict the solution proposed to solve this situation, the first prototype of the Content Personalization service (CP), a module that selects the most suitable resource for a specific student taking into account preferences, device properties and resource properties together with the adaptations available.

1. Content Personalization

The main goal of EU4ALL project is to design and implement an extensible architecture of services to support accessible lifelong learning for adult learners with special needs. These services are to be open, secure, standard-based, accessible and interoperable.

The CP is a component in the EU4ALL project. It is a resource selection module. The goal of the CP is find the most suitable resource resolving a request coming from a Virtual Learning Environment (VLE) and returning an identifier for the chosen resource. This way, the module provides users with the educational resource that best adapts to the preferences and needs of the student. To choose the best resource, the CP processes the user preferences (stored in the user profile), the device preferences (stored in the device profile) and the characteristics of the requested resource and its available adaptations (resource profile/adaptation).

The standards used to implement the profiles are:

- W3C CC/PP [7] (Composite Capability/Preference Profiles): Specification that defines the capabilities of the devices that the user is using. CC/PP is based on RDF (Resource Description Framework). In the EU4ALL project, we are using UAProf [8] (User Agent Profile) specification, based on the CC/PP standard. UAProf is concerned with capturing capability and preference information for devices, and this information can be used by content providers to produce content in an appropriate format for the specific device.
- IMS AccLIP [9] (Accessibility for Learner Information Package): AccLIP documents only store accessibility preferences of the user. It could be integrated with the standard IMS LIP [10] (Learning Information Package).
- ISO DRD [11] (Digital Resource Description, ISO/IEC 24751-3:2008): New standard use to describe the learning objects, both original resources and adaptations. ISO DRD is the 3rd part of the *Individualized adaptability and accessibility in e-learning, education and training specification ("Access for all")*. It provides a common language for describing digital learning resources to facilitate matching of those resources to learners' accessibility needs and preferences.

2. Content Personalization behavior

The behavior of the CP module is summarized in the figure 1.

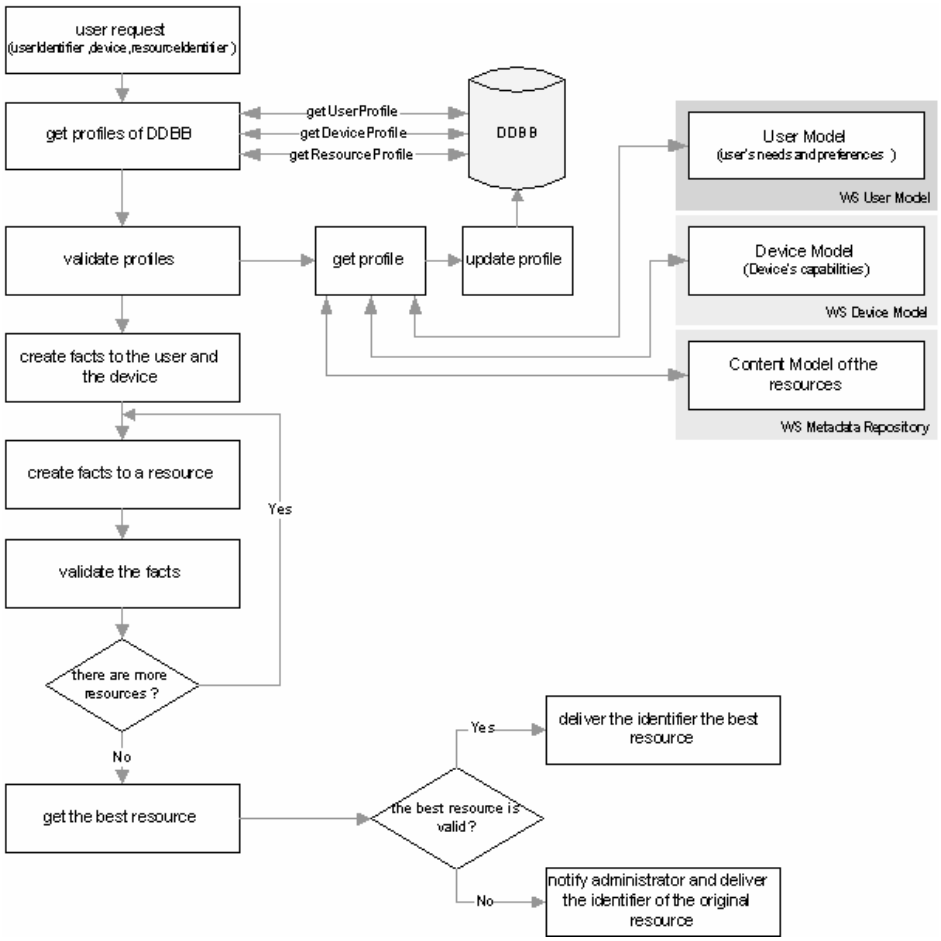

Figure 1. Flow diagram of the CP module

When the CP modules receives a request with the user identifier, a short description of the device and the resource identifier, the CP retrieves the profiles for the user, the device the resource and its available adaptations. There are two alternatives to retrieve the profiles, the first one to look for the profiles in the cache memory of the CP module and the second is to access the external modeling services (User modeling, Device modeling) via web services. In the first place, the CP module tries to retrieve the profiles from the cache, if these profiles are invalid or there are no profiles available, it tries to retrieve the profiles using external modeling services and it updates the cache with them. An invalid profile is a profile that has exceeded the timeout specified. The threshold for the timeout is configurable.

When all the profiles retrieved by the CP are valid, it creates the facts for the user and device profiles. The facts are implemented in an intermediate language understood by the rule based engine. The profiles of the original resource and its adaptations are translated into a group of facts, where each group has the facts that correspond to a resource or to an adaptation. Then the CP loads the facts for the user, the device and the resource/adaptation in the rule based engine, and it validates them getting a score for

each resources. This score is stored together with the identifier of the analyzed resource. This operation is repeated until all the adaptations have been analyzed. Once the adaptations have been analyzed, if the resource/adaptation with the highest store is above a certain threshold, this will be marked as the most suitable resource for the user, as result, the CP returns the identifier of this resource. On the contrary, if the value for the highest score does not pass the threshold, the CP returns an identifier for the original resource and sends an e-mail to the administrator to notify that there is no adaptation available for a certain user and device.

3. Architectural Components of the Content Personalization Module

The CP module is made up of different components with the structure indicated in the figure 2.

3.1. Web Services Layer

The layer contains the available web services of the CP, and also it is used to access the external modeling services to obtain the profiles of the user, the device, the resources and the adaptations. By using web services it is possible to distribute the tasks among different modules that are completely independent without having to share features such as the programming language or the operative system. The first prototype for the CP uses Axis2 1.3 [12] for SOAP.

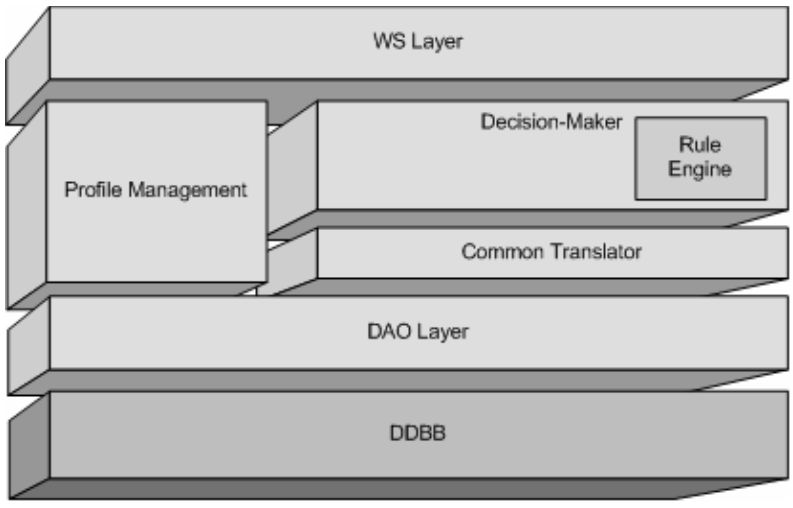

Figure 2. Content Personalization components

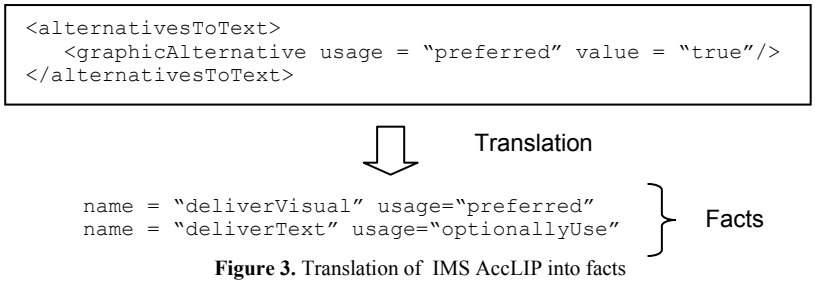

Figure 3. Translation of IMS AccLIP into facts

3.2. Common Translator

This component is in charge of translating the profiles to facts. These facts are implemented in an intermediate language managed by Decision-Maker. It will interpret the facts in order to validate the different resources and the adaptations. This way, the CP is standard independent by using Decision-Maker, since it works with an intermediate language. This eases the integration of new standards with the CP. To manage a new standard only implies to modify the Common Translator module, providing it with the capability to translate the new Standard into the intermediate language. The Common Translator component uses Jena 2.5.6 [14] libraries, which is a framework for semantic web that is also used to manage CC/PP standard through RDF. Figure 3 shows a translation example of a fragment of IMS AccLIP into facts. Of one user prefers a graphical alternative over a textual resource, this is translated to two facts; the first one is to deliver the visual resource as the preferred one and the second is to deliver the textual resource as the optional one. The values of the usage fields in the facts show that the user prefers a visual resource, but if it does not exist he accepts a textual resource. These facts mean the same that the piece of IMSAccLIP where the user prefers the visual resource, but he does not require it. Note that the *"usage"* attribute is setting the level of preference in the IMS AccLIP and could be *"preferred"*, *"optionallyUse"*, *"required"* or *"notUse"*.

3.3. Decision-Maker

This component is in charge of validates the facts. The first prototype for the CP module uses Drools [15] as rule based engine. The module provides a simple way to take AI decisions. The following code features an example for a rule that decides is a resource is valid. In the first prototype of the CP, a resource is valid if the visual, textual, auditory, format and language bits are valid. The rule `"Visual missing"` checks id the resource is not visual and the user required a visual resource. The rule `"Hardware visual ok"` checks if the device is compliant with the hardware handed to display a visual resource. `"User visual ok"` checks if the user wants a visual resource, and `"Visual ok"` checks if the hardware can handle a visual resource and if it is accepted by the user. Finally, the rule `"Visual ko"` is the oposite operation to `"Visual ok"` and decides when the user does not want a particular resource or when the hardware is not appropriate.

```
rule "Visual missing"
  no-loop true
  when
    not(exists(ResourceCharacteristic(name==“accessMode”, value==“visual”)))
    UserPreference(name==“deliverVisual”, usage!=“required”)
    $mUserVisualOk:Message(message==“userVisualOK”)
    $mHardwareVisualOk:Message(message==“hardwareVisualOk”)
  then
    $mUserVisualOk.setValue(true);
    update($mUserVisualOk);
    $mHardwareVisualOk.setValue(true);
    update($mHardwareVisualOk);
end

rule "User visual ok"
  no-loop true
  when
    ResourceCharacteristic(name==“accessMode”, value==“visual”)
    UserPreference(name==“deliverVisual”, $usage:usage!=“notUse”)
    $mUserVisualOk:Message(message==“userVisualOk”)
  then
    $mUserVisualOk.setValue(true);
    $mUserVisualOk.setUsage($usage);
    update($mUserVisualOk);
end

rule "Hardware visual ok"
  no-loop true
  when
    ResourceCharacteristic(name==“accessMode”, value==“visual”)
    DeviceCharacteristic(name==“imageCapable”,
                         value=="Yes"||value==null)
    $mHardwareVisualOk:Message(message==“hardwareVisualOk”)
  then
    $mHardwareVisualOk.setValue(true);
    update($mHardwareVisualOk);
end

rule "Visual ok"
  no-loop true
  when
    Message(message==“userVisualOk”, value==true, $usage:usage)
    Message(message==“hardwareVisualOk”, value==true )
    $mVisualOk:Message(message==“visualOk” )
  then
    $mVisualOk.setValue(true);
    $mVisualOk.setUsage($usage);
    update($mVisualOk);
end

rule "Visual ko"
  no-loop true
  when
    (not
      Message(message==“userVisualOk”, value==true)
      and
      Message(message==“hardwareVisualOk”, value == true)
    )
    $mVisualOk:Message(message==“visualOk”)
  then
    $mVisualOk.setValue(false);
    update($mVisualOk);
end
```

3.4. Profile Management

It works with all the profiles as if they belonged to a single type with regards to the standard. It retrieves the profiles from the cache using the DAO's and decides if they are valid. If they are not valid, it is interacts with the Web Services layer to retrieve the profiles and launch the update process for the cache.

3.5. Data Access Object Layer

It abstracts and encapsulates all the access to the data. DAO [13] layer administers the connection with the data source to retrieve and store data, hiding implementation details inside the DAO's. It paves the way to the possible migration to a different source of data, since in this case only the DAO layer needs to be modified. In addition, it centralizes all the accesses to data in a separate layer.

4. Integration of the CP Module with others EU4ALL and VLE Components

The integration with the EU4ALL modules is based on SOA (Service Oriented Architecture) [6], see Figure 4. This way an open architecture is achieved, easing the changes of processes and the integration of different technologies. This is very important since the functionality of the CP can be invoked from different VLE's and these can be programmed in different programming languages such as Tcl (.LRN), PHP (Moodle), etc … On the other hand, the CP needs to connect to other modules such as User Modeling, Device Modeling and Metadata Repository in a transparent way. The communication protocol used by the CP with the VLE's and the other modules is SOAP (Simple Object Access Protocol). The SOAP protocol wraps the different standards used in the communication. For instance, User Modeling responses to CP with IMS-AccLIP, on the other hand, Device Modeling uses CC/PP in the response, finally, Metadata Repository uses ISO-DRD in the communication with CP. All these standards are understood by the CP and they are translated to facts.

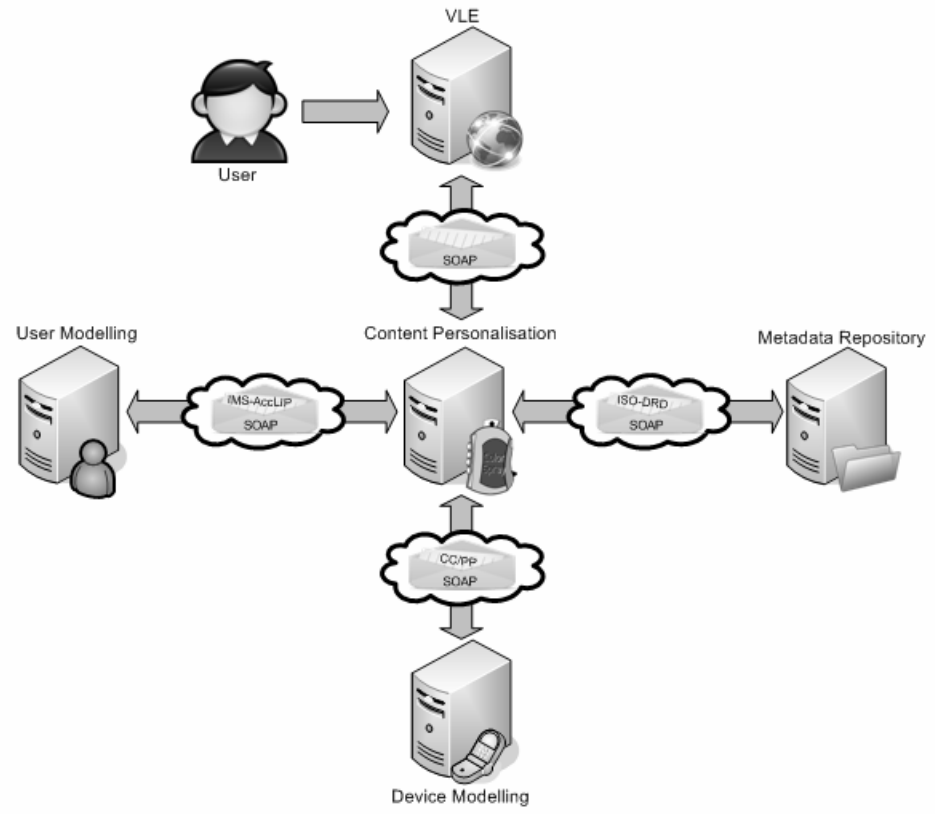

Figure 4. Connection schema between CP and the other modules

5. Conclusions and Future Work

The first prototype of the CP has been implemented with Java 1.5 and allocated in an Apache 5.5.23 web server. An open modular architecture has been created to easily change the profile retrieval, the cache storage and the decision make algorithm using different design patters. The module has been developed using SOA architecture, and as a result the different modules of the architecture are totally independent.

Using a rules based engine in the Decision-Maker component eases the comprehension of the decision making process of the CP and enables to modify the business logic without having to rebuild the module. The structure of the CP allows changing the rules based engine easily.

A graphical user interface (GUI) has been developed for testing purposes of the Expert System. The interface allows creating profiles (users, devices, resources) and to request a personalization for a resource. The figure 5 shows the main page of the GUI. Currently the prototype is under evaluation and the results will be available in late 2009.

A possible root to investigate would be to integrate new standards to the ones already managed by the CP like ISO PNP [16] (Personal needs and Preferences, ISO/IEC 24751-2:2008) which is used to model user preferences and whose integration with ISO DRD is better than the one provided by IMS AccLIP. For the second prototype of the CP it could be possible to conduct a study of the different AI techniques to be used to enhance the performance of the or to substitute the rules based engine with another alternative.

The CP module integrated within EU4ALL architecture will be evaluated on a large scale in European universities at the beginning of 2010. That year the evaluation outcomes will be disseminated and could be checked whether the use of the component is really useful for the students taking into account their needs and preferences.

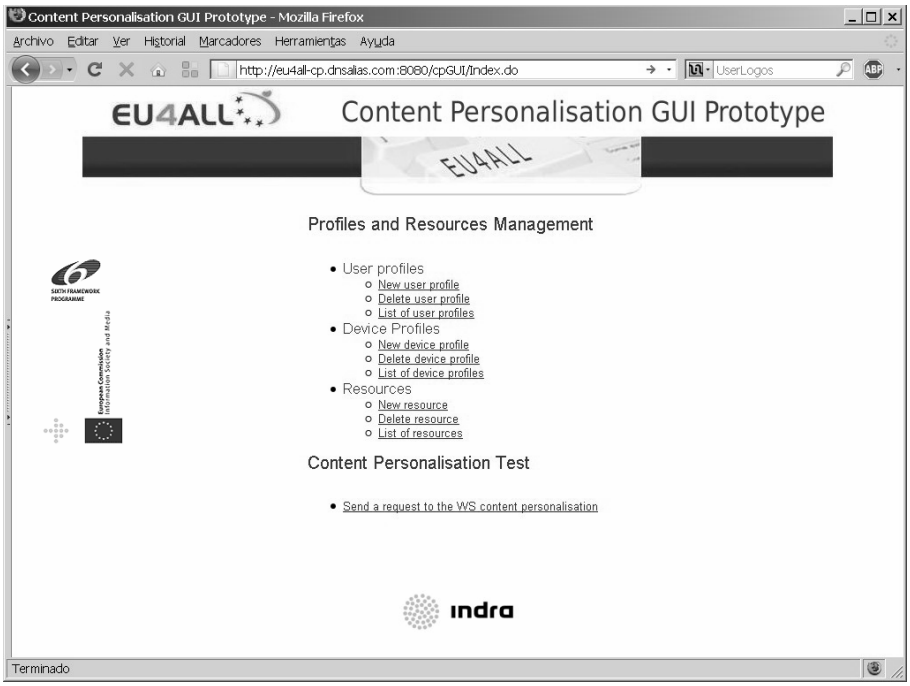

Figure 5. Main page of the GUI for the Content Personalization module

Acknowledgements

The authors would like to thank the European Commission for partly funding EU4ALL project [17] (European Unified Approach for Assisted Life Long Learning - IST-FP6-034778).

References

[1] Pérez de la Cámara S., Fernández del Viso A., Rebate C., Rubión E., Vázquez de Prada E., Gómez N., *Content Personalization: assembling the puzzle of standards*. IADIS Multi Conference on Computer Science and Information Systems (2008), e-Learning Volume II. ISBN: 978-972-8924-58-4, pp. 143-146.

[2] Brusilovsky P., Kobsa A., Nejdl W., The *Adaptive Web: Methods and Strategies of Web Personalization*, Volume 4321/2007, Springer Berlin, ISBN 978-3-540-72078-2, 2007.

[3] Velasco C., Mohamad Y., Carmien S., *Universally Accessible Lifelong Learning by User and Device Profiling Adaptation*, TUMAS-A workshop at User Modelling Conference (2007), pp. 23-26.

[4] Karpouzis K., Moschovitis G., Ntalianis K., Ioannou S., Kollias S., *Web Access to Large Audiovisual Assets Based on User Preferences*, Multimedia Tools and Applications, Vol. 22 Issue 3, Kluwer Academic Publishers, 2004.

[5] Rousseau B., Browne P., Malone P., ÓFoghlú M., *User Profiling for Content Personalisation in Information Retrieval*, 19 Congress ACM SAC (2004).

[6] Newcomer E., and G. Lomow. *Understanding SOA with Web Services*, Addison-Wesley Professional, ISBN: 978-0321180865, 2004.

[7] W3C CC/PP Information Page: http://www.w3.org/Mobile/CCPP/

[8] UAProf in Open Mobile Alliance: http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf

[9] IMS AccLIP specification: http://www.imsglobal.org/accessibility/acclipv1p0/imsacclip_infov1p0.html

[10] IMS LIP Specification: http://www.imsglobal.org/profiles/

[11] ISO "Access for all" DRD: http://www.iso.org/iso/catalogue_detail?csnumber=43604

[12] Official page of Axis2: http://ws.apache.org/axis2/

[13] Data Access Object description provided by Sun Microsystems: http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

[14] Official page of Jena: http://jena.sourceforge.net/

[15] Official page of Drools: http://www.jboss.org/drools/

[16] "Access for all" PNP: http://www.iso.org/iso/catalogue_detail?csnumber=43603

[17] Official page of the project EU4ALL: http://www.eu4all-project.eu/